

CLAIMS

What is claimed is:

Claim 1. A method of compiling code executable in an electronic device having a component architecture platform (CAP) framework, the method comprising:

- compiling code targeted to the CAP framework;
- creating a reference lookup table; and
- saving compiled code and the reference lookup table.

Claim 2. The method according to claim 1, wherein compiling code comprises:

- generating program instructions that use symbolic references for at least one of program variables, functions, methods, and operands; and
- extracting reference information and populating the reference lookup table with extracted reference information.

Claim 3. The method according to claim 2, wherein creating a reference lookup table comprises:

- associating at least one symbolic reference with a physical address; and
- maintaining the reference information in entries, wherein the entries map symbolic references encountered in the code during compilation to memory addresses.

Claim 4. The method according to claim 3, wherein the entries comprise address values assigned to references corresponding to one of a compiler phase and a link/load phase of the compilation.

Claim 5. The method according to claim 1, further comprising generating one of binary and hex output for compiled code that create entries corresponding to at least one entry in the reference lookup table.

Claim 6. The method according to claim 1, wherein the compiled code is adapted for executing individual program instructions, and the individual program instructions are pre-fetched into a pipeline of instructions in preparation for execution.

Claim 7. The method according to claim 6, wherein processing time is reduced by resolving references via a reference lookup table management unit, and the unit facilitates resolution of the references for pre-fetching instructions into the pipeline.

Claim 8. The method according to claim 1, wherein the reference lookup table comprises mapping of symbolic references encountered in code during compilation to memory addresses.

Claim 9. The method according to claim 1, wherein the reference lookup table comprises symbolic references and associated address or offset values.

Claim 10. A method of generating update packages for updating software in a mobile electronic device capable of employing the component architecture platform (CAP) framework, the method comprising:

retrieving an existing and an updated version of code;

determining which program components to modify;

generating an update package having modules corresponding to those program components to be modified; and

generating an associated reference lookup table having entries corresponding to those program components to be modified.

Claim 11. The method according to claim 10, wherein determining which program components to modify comprises determining program components to be one of left unchanged, deleted, added, and modified, and wherein program components left unchanged are not included in the update package.

Claim 12. The method according to claim 10, wherein the update package and the associated reference lookup table modifications are adapted for transfer to an embedded system in the electronic device as one of a single program unit and two different related program units transferred when the electronic device is updated.

Claim 13. The method of claim 10, wherein the update package is adapted to facilitate update by an update agent in the electronic device.

Claim 14. The method of claim 10, wherein the update package comprises information for adding new modules, and further comprising:

- replacing existing modules with new modules; and
- updating the associated reference lookup table.

Claim 15. A method of operating a processor supporting a component architecture platform (CAP) framework, the method comprising:

- determining an initial processor execution mode;
- setting the initial processor execution mode;
- initializing with a reference lookup table management unit ; and
- executing program instructions.

Claim 16 The method according to claim 15 further comprising:

- looking up additional references; and
- populating a pipeline with at least one of program instructions, references, and addresses.

Claim 17 The method according to claim 15 further comprising:

- initializing a reference lookup table with symbolic reference values to be resolved; and
- retrieving addresses for information corresponding to the symbolic reference values and resolving them prior to execution.

Claim 18. A mobile electronic device having an embedded system capable of employing a component architecture platform (CAP) framework, the device comprising:

- a processor;
- a reference lookup table management unit;
- primary memory and secondary memory; and
- at least one reference lookup table.

Claim 19. The device according to claim 18, further comprising:

- non-volatile memory;
- volatile memory; and
- software resident in at least a portion of primary and secondary memory.

Claim 20. The device according to claim 18, wherein the processor retrieves program instructions, resolves symbolic references employing the at least one reference lookup table and executes program instructions retrieved from at least one of primary and secondary memory.

Claim 21. The device according to claim 20, wherein executing a program instruction comprises performing at least one reference lookup, the at least one reference lookup using symbolic names for one of operands, modules, method names, functions, and components, and wherein the at least one reference lookup may be resolvable into an address at program runtime.

Claim 22. The device according to claim 18, wherein at least one of primary and secondary memory comprises FLASH memory.

Claim 23. The device according to claim 18, wherein the at least one reference lookup table comprises a local reference lookup table mapping at least one of local variables, functions, and methods.

Claim 24. The device according to claim 23, wherein modifications to program instructions are reflected and recorded in the local reference lookup table.

Claim 25. The device according to claim 24, wherein modifications reflected in the local reference lookup table are also reflected in a non-local reference lookup table.

Claim 26. A computing environment with a processor capable of executing program instructions from a pipeline comprising:

a reference lookup table that is capable of being populated with symbolic references and corresponding location values; and

a reference lookup table management unit that is capable of managing the reference lookup table, and resolving symbolic references in the program instructions into location values before population into the pipeline, wherein the processor is capable of retrieving the program instructions and resolved symbolic references and executing the program instructions.

Claim 27. The computing environment according to claim 26, wherein location values in the reference lookup table are one of memory addresses and offsets.